

Exercises

- Define a class called **Logger** that takes the name of a file in its constructor and provides a method for writing messages to that file. Don't worry about the actual file I/O operations. Just define the class with the right members.
- Given the **Person** class below, create a getter for getting the full name of a person.

```
class Person {  
    constructor(public firstName: string, public lastName: string) {}  
}
```

- Create a new class called **Employee** that extends **Person** and adds a new property called **salary**.
- What is the difference between **private** and **protected** members?
- Given the data below, define an interface for representing employees:

```
let employee = {  
    name: 'John Smith',  
    salary: 50_000,  
    address: {  
        street: 'Flinders st',  
        city: 'Melbourne',  
        zipCode: 3144,  
    },  
};
```

Solutions

- Define a class called **Logger** that takes the name of a file in its constructor and provides a method for writing messages to that file. Don't worry about the actual file I/O operations. Just define the class with the right members.

```
class Logger {  
    constructor(public logFile: string) {}  
  
    log(message: string) {}  
}
```

- Given the **Person** class below, create a getter for getting the full name of a person.

```
class Person {  
    constructor(public firstName: string, public lastName: string) {}  
  
    get fullName() {  
        return `${this.firstName} ${this.lastName}`;  
    }  
}
```

- Create a new class called **Employee** that extends **Person** and adds a new property called **salary**.

```
class Employee extends Person {  
    constructor(  
        firstName: string,  
        lastName: string,  
        public salary: number) {  
        super(firstName, lastName);  
    }  
}
```

- What is the difference between **private** and **protected** members?

Private members are not inherited by child classes.

- Given the data below, define an interface for representing employees:

```
interface Address {  
  street: string;  
  city: string;  
  zipCode: number;  
}
```

```
interface Employee {  
  name: string;  
  salary: number;  
  address: Address;  
}
```